REMARKS

I. INTRODUCTION

Claims 6 and 19 have been cancelled. Claims 1-5, 7-13, 20, 26-35, and 45-51 have been amended. Claims 1-5, 7-18, and 20-55 remain pending in the present application. No new matter has been added. In view of the above amendments and the following remarks, it is respectfully submitted that all of the presently pending claims are allowable.

II. THE 35 U.S.C. § 101 REJECTIONS SHOULD BE WITHDRAWN

Claims 1-44, and 47-55 stand rejected under 35 U.S.C. § 101. Applicants have amended the claims to produce a useful result when fixed in a tangible medium. Thus, it is respectfully submitted that the rejections be withdrawn.

III. THE 35 U.S.C. § 102(b) REJECTIONS SHOULD BE WITHDRAWN

Claims 1-5, 13-18, 26-28, 35-38, 45, and 46 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,339,435 to Lubkin et al. (hereinafter "Lubkin"). At the Examiner's suggestion, Applicants have amended the claims to include domain hierarchy and to clarify the term "symbol" in order to further distinguish the present invention from Lubkin.

Lubkin generally describes a heterogeneous management tool, which resides on one computer in a network of heterogeneous computers. This tool enables a user to build software systems on the resident computer or on another computer in the network supporting the same or different type of binaries. (*See* Lubkin, col. 3, ll. 61-67). A system build begins with a user specifying a system model and configuration threads on the host computer. (*See* Id., col. 6, ll. 24-26). The system model describes the structure and specifies the components of the desired software system, along with the interrelationship of the components and their attributes. (*See* Id., col. 6, ll. 28-31). The configurations threads are user-provided descriptions of the versions and translator options desired to be used during building of each component of the system model. (*See* Id., col. 7, ll. 35-38). The management tool combines the system model and configuration thread to make a desired bound configuration thread (BCT) for each component. (*See* Id., col. 10, ll. 48-68). The management tool then compares the BCTs of the desired proposed system to the BCTs existing in the system to determine whether and which of any of the components of the specified desired software system have already been built. (*See* Id., col. 10, ll. 48-68). If the management tool determines that building is required, it determines which building tools are necessary and selects a foreign computer to build the system. (*See* Id., col. 11, ll. 21-53). If the selected builder computer is unauthorized, inaccessible, or improperly configured, then the management tool marks it as such and selects a new builder computer. (*See* Id., col. 13, ll. 40-56).

In response to a user input build command, this management tool makes a bound configuration thread (BCT) for each component of the proposed system by combining the system model with the configuration thread. (*See* Id., col. 10, ll. 50-56). The BCT is a permanent record of the options, source versions and tool version used to build the respective component and other characteristics of the build context. (*See* Id., col. 10, ll. 56-59). User binary pools are special directories where BCTs and derived objects of previous builds are cached. (*See* Id., col. 10, ll.

60-63). The management tool will look into these special directories containing permanent records of the options, source versions and tool version that were used to build the build context and compare the existing record to the record of options, source versions, and tools versions of the desired proposed system. (*See* Id., col. 10, ll. 63-65). From this comparison, the management tool will determine which of the components of the desired proposed system has been built. (*See* Id., col. 10, ll. 65-68). If the binary (build results) of a component is not found in a pool, then the management tool determines that a build is required for the component. (*See* Id., col. 11, ll. 21-23). The management tool arranges a sequence of builds, one build for each such component determined to require building, and determines which building tools are necessary. (*See* Id., col. 11, ll. 23-27). For each component that must be built, the management tool will look to the host type specified for the component in the system model. (*See* Id., col. 11, ll. 27-29).

The present invention relates to a method for determining whether an application program or an operating system within a computing environment of a system project uses components that are not already included within the system project. These needed components are shown to the user of the project facility, who may then select the components to be added to the system project. Alternatively, these needed components may be automatically added to the system project.

Claim 1 recites "determining a set of symbols imported by the set of modules assigned to the domain, wherein each symbol identifies a memory location storing one of a code and a data structure." The Examiner contends that this limitation is anticipated by Lubkin's selection of a foreign builder computer for each component which is required to be built in order to satisfy the build command input by the user. (*See* Office Action, p. 2, ¶ 4). As described in an exemplary embodiment of the present invention, compiled source code ("modules") are fed to an examination utility, which reads each module and identifies any symbols in the module as either exported or imported by the module. (*See* Specification, p. 14, ll.19-21). A "symbol" is a name that represents a memory location of a code or data structure. (*See* Specification, p. 15, ll. 6-10). Claim 1 has been amended to specifically recite this limitation. Thus, the symbols which are identified in this process, point to other sources of code or data. In contrast, the foreign builder computer is not an identifier of sources of code, but is a tool used to construct the user-specified system. The selection of these tools is not equivalent to searching for and locating symbols within a module.

The Examiner states in the Response to Arguments that the foreign builder will look for a build list on another computer if the list is not found. (*See* Office Action, p.11, ¶ 2). The build list file is not equivalent to a set of symbols within a module, but rather they are a list of candidate foreign builders. Thus, Applicants maintain that Lubkin's disclosure neither teaches nor suggests "determining a set of symbols imported by the set of modules assigned to the domain, wherein each symbol identifies a memory location storing one of a code and a data structure" as recited in claim 1, and the rejection of claim 1 should be withdrawn.

As claims 2-5 depend from, and therefore include all the limitations of claim 1, it is hereby submitted that these claims are also allowable.

The Examiner rejected claim 13 on the same grounds as claim 1, indicating that claim 13 was merely the system version of claim 1. (*See* Office Action, p. 3) The Examiner used the same rationale to reject claim 45, indicating that claim 45 was merely the device version of claim 1. (*Id.,* at p. 5) For the reasons stated above with respect to claim 1, Applicants respectfully submit that claims 13 and 45 are also allowable and request that the rejections be withdrawn. As claims 14-18 depend from and therefore include all the limitations of claim 13, it is submitted that these claim are also allowable.

The Examiner rejected claim 26 as being anticipated by the disclosure of Lubkin. (*See* Office Action, p. 4) Claim 26 has been amended to include the same recitation as claim 1. Thus, Applicants respectfully submit that for at least the reasons discussed above with respect to claim 1, claim 26 should also be allowed. Therefore, Applicants request that the rejection of claim 26, and the rejections of claims 27-28, which depend from and include all the limitations of claim 26, be withdrawn.

The Examiner rejected claims 35 and 46 under the same rationale as claim 26. The Examiner indicated that claim 35 is merely the system version of claim 26, and that claim 46 is the device version of claim 26. (*See* Office Action, p. 5) Therefore, Applicants respectfully request that for at least the reasons states above with respect to claim 26, the rejections of claims 35 and 46 be withdrawn. Because claims 36 - 38 depend from and therefore include all the limitations of claim 35, Applicants request that these claims also be allowed.

IV. THE 35 U.S.C. § 103(a) REJECTIONS SHOULD BE WITHDRAWN

The Examiner has rejected claims 6-12, 19-25, 29-34, and 39-44 under 35 U.S.C. § 103(a) as unpatentable over Lubkin in view of U.S. Patent No. 5,528,757 to Yamasaki (hereinafter "Yamasaki"). (*See* Office Action, pp. 5-9) As discussed above, Lubkin does not teach or suggest all the limitations of independent claims 1, 13, 26, 35, 45, and 46. Yamasaki does not cure these defects of Lubkin. Accordingly, because claims 6-12, 19-25, 29-34, and 39-

44 depend from and, therefore, include all of the limitations of corresponding claims 1, 13, 26, 35, 45, and 46, it is respectfully submitted that these claims are also allowable over the cited references.

CONCLUSION

In light of the foregoing, Applicants respectfully submit that all of the now pending claims are in condition for allowance. All issues raised by the Examiner having been addressed. An early and favorable action on the merits is earnestly solicited.

Respectfully submitted,

Dated: September 2, 2005

By:______________________
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038